UNIVERSITY OF OULU
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Nuutti Räihä
Eetu Huusko

A Visualization Service for Parkinson's Disease Motoric Measurements

Bachelor's Thesis
Degree Programme in Computer Science and Engineering
May 2019

Räihä N., Huusko E. (2019) A Visualization Service for Parkinson's Disease Motoric Measurements. University of Oulu, Degree Programme in Computer Science and Engineering, 44 p.

ABSTRACT

In the era of affordable sensors and smart devices, medical self measurements are increasingly common. Currently, powerful mobile phones and different sensor systems facilitate a variety of use cases for patients, enthusiasts and experts alike to measure their bodily condition. Visualizing the results of these measurements is not always a certainty, as the data could be complex or difficult to interpret in some other way.

We design and develop a web application to visualize measurement data collected with an existing self-measurement application called STOP. STOP collects acceleration and gyroscopic data about hand tremor with parkinson's disease patients. Our application, Medidash, draws dynamic temporal cartesian graphs from these measurements. Medidash consists of an existing AWARE-database and a Python-Dash based web application.

Keywords: Medical visualization, web application, data visualization, user interface.

Räihä N., Huusko E. (2019) Visualisointipalvelu Parkinsonin Taudin Oireiden Mittausten Kuvaamiseen. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 44 s.

TIIVISTELMÄ

Edullisen kuluttajaelektroniikan ja älylaitteiden aikakaudella lääketieteellinen mittaaminen yleistyy voimakkaasti. Nykyiset tehokkaat matkapuhelimet ja erilaiset sensorijärjestelmät mahdollistavat erilaisia käyttötarkoituksia terveyden mittaamiseen niin potilaille, ammattilaisille kuin harrastajillekin. Näiden mittausten visualisointi ei aina ole itsestäänselvyys, sillä kerätty data saattaa olla monimutkaista tai muuten hankalasti tulkittavaa.

Suunnitelemme ja kehitämme verkkosovelluksen jo olemassa olevalla kehonmittausta tekevällä STOP-applikaatiolla kerätyn datan visualisointiin. STOP kerää kiihtyvyysanturilla ja gyroskoopilla dataa Parkinsonin taudin potilaiden käsien tärinästä. Kehittämämme sovellus, Medidash, piirtää dynaamisia kuvaajia mitatun datan visualisoimiseksi. Medidash koostuu jo olemassaolevasta AWARE-tietokannasta sekä Python-Dash -pohjaisesta verkkoapplikaatiosta.

Avainsanat: Lääketieteellinen visualisointi, visualisointi, verkkoapplikaatio, käyttöliittymä.

TABLE OF CONTENTS

FOREWORD

We would like to thank all the faculty staff that has helped us with this project, especially Elina, Valerii, Simo and Matti. Special thanks to our families for all their love and support, and the café in front of lecture hall L2 for hosting most of our team meetings.

Oulu, 19th May, 2019

Nuutti Räihä, Eetu Huusko

LIST OF ABBREVIATIONS AND SYMBOLS

UI	User interface
PD	Parkinson's disease
ECG	Electrocardiogram
UX	User experience
REST	Representational state transfer
HTTP	HyperText Transfer Protocol
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DG	Design guidelines
API	Application Programming Interface
JSON	JavaScript Object Notation
URI	Uniform Resource Identifier
DOM	Document Object Model
URL	Uniform Resource Locator
SQL	Sequential Query Language
MVC	Model View Controller
SUS	system usability survey
XSS	Cross-site Scripting
R	triangular matrix
Q	orthogonal matrix
X	model matrix
y	response variable
N	total amount of usability problems
L	proportion of usability problems discovered while testing a single user
$\hat{Y}$	fitted values
β	least squares parameters
$\hat{\beta}$	estimated least squares parameters
n	number of samples

1. INTRODUCTION

As the need for personally tailored medical information increases, applications for patients, doctors and other health care workers are being developed. Many applications have already been developed for a wide range of use cases. The applications may range all the way from measuring the sleep quality nightly, to measuring blood insulin levels or overall activity level. Different applications require either active or passive involvement from the patient, depending on whether the patient has to actively input data to the application or if the application measures data independent of the patients actions. Overall, these applications aim to increase the awareness of the user. [1, 2]

Many applications that collect medical data also present a view to the data. These views can be offered on an external device, personal mobile phone or even in a web browser. The difficulty of developing a dashboard for medical information visualization is increased as the motoric and visual capabilities of the end user decline. This means that if the user has limited ability to control his or her movements, the requirements of the user interface change accordingly. The ease of use, clarity and unambiguous controls are highlighted in this kind of development.

In this document we introduce the concept of distributed web-services and visualizing medical time series data. We clarify system requirements and then design and implement a service for visualizing spatio-temporal measurement data gathered from Parkinson's disease patients. Visualization is aimed at two different user groups; Parkinson's disease patients and their doctors. The data collection software is the fruit of a previous study, which we build our project on top of. First, we look at previous work and technologies utilized in our implementation in chapter 2. After implementation in chapter 3, evaluation is done with real users in chapter 4. Finally, we talk about project results and further development possibilities in chapter 5.

1.1. Contribution

This project was completed in the Applied Computer Project 1 -course at University of Oulu. The project's workload was shared according to each person's strengths and weaknesses. The work consisted of researching, designing (the user interface and the way of implementation), actual implementation, evaluation and writing the report and the additional presentations for the course. Eetu Huusko and Nuutti Räihä did most of the studies about Parkinson's disease and designing the user interface for their limitations as well as preparing the thesis as the key output from this project. Tuomas Varanka planned and designed most of the mathematical sections for the analytics in the user interface and Atte Jauhiainen was responsible for the technical implementation and making the application run in a secure environment in the internet.

Research was also on-going through the whole project. A lot of work was done together, mainly designing and making the important decision about the project's direction.

2. RELATED WORK AND TECHNOLOGIES

As the era of stationary personal computers is coming to an end, and we're progressing towards ubiquitous computing, it is easier and easier to develop mobile applications. This trend has opened the door for different personal health care applications to be developed. It is predicted that in the future, it will be increasingly common for an individual to keep track of some health related attributes [1, 3].

As the information technology grows rapidly and the health industry starts to take the advantage of it, it creates more pressure for individual patients to use new technologies to keep track of the health related attributes. More than often people who would gain the most from logging the data are elderly and people who suffer from long-term illness. Unfortunately the rapid growth can feel overwhelming for elderly people, as technology evolves faster than they can learn to use it. User interface design and user experience design become more important so people with limitations or disabilities from illnesses or little knowledge of technology can keep up with applications they need to use.

Data visualization is constantly moving towards a more important role, given the increasing amount of data. Finding trends and relationships from the raw data, especially in the case of big data can be very challenging for human beings. Visualization tools give the opportunity to look at large amounts of data and find trends and relationships between the data with a quick glance. In medical data and especially with long term illnesses correlations between variables over potentially long periods of time are the key. [4]

Numerous visualization tools for presenting medical data have been developed over the years for both clinicians and medical professionals and their differences have been evaluated in [4, 5]. The authors asked feedback and suggestiongs from both clinicians and students for the features of visualization tools. Based on their feedback the main features for a visualization tool should be ease of use, interactivity and multitasking (viewing multiple plots at once). With ease of use the evaluators want simplicity and a quick learning curve without sacrificing the ability for more advanced methods. Designing the UI (User Interface) has a tremendous effect on the usability of the tool. Interactivity of the tool can consist of the following: zooming and panning of the graph, ability to change location of different plots, ability to see values of individual or chosen points and the ability to compare chosen values.

Patients with chronic diseases may benefit from visualized data, and being able to compare it to reference data. People who participated in using PatientsLikeMe [6] and answered the cross-sectional online surveys found that sharing and discussing about their health information online helped them to learn more about their symptoms they had been experiencing, understanding better how their treatments were working and felt more involved in the treatment decision. The findings were in line with national consumer survey on health information technology, which found that benefits included feeling that the patients knew more about their health, feeling like they knew more about the care they were being given by their doctor and feeling able to ask their doctor questions they would not have asked otherwise.

In Parkinson's disease, kinematic features are used to evaluate symptom severity and motor complications in order to optimize the treatment dosages [7]. With different data visualization techniques, such as time-plotted, and animations, we can deliver this information. With data visualization, clinicians can gain more detailed insight of treatment-related motor complications. Visualization helps to recognize motor dysfunction patterns and visualized motor tests may be useful when evaluating therapy-related complications, such as under- and over-medications in PD (Parkinson's disease).

In a study conducted by The University of British Columbia and Indian Institute of Technology Madras [8], researchers examined the possibility of using wireless wearable sensors to recognize Parkinson's disease patients onset of involuntary tremors. The accelerometer and gyroscope sensors output were plotted as a three-dimensional graph after applying a dimensionality reduction technique, principal component analysis, on the data. This was mainly done as a visual aid to help the development of the prototype in question.

Visualization applications have been created for tracking other diseases as well. In [9], the authors developed an open source visualization library for python for visualizing ECG data from patients suffering from Long QT Syndrome. Portable Holter monitors were used to monitor the patients for up to 24 hours since a typical short time series of ECG will not reveal any "interesting" cardiac events. A single 24 hour recording from the Holter recordings consisted of approximately 120,000 QTc points. Plotting this many data points into a single plot clusters the data and makes it hard to see. In addition to the inevitable outliers and noise a median filter with a 10 minute window was used to solve both the mentioned problems. The median filter smooths the noise while also reducing the amount of data points greatly. Long QTc times can be detected in the night when the patient is a sleep—thus creating a cyclic event occurring daily. A cyclic representation for visualization was thus chosen. Analyzing the data in a typical Cartesian axis created problems with discontinuities as the morning and night were far away from each other. Thus the authors ended up using polar coordinates with a 24 hour clock. The cyclic polar axis representation allowed to plot the data in a natural and familiar looking clock where the trend of increased QTc times can be viewed with a glance.

2.1. Parkinson's Disease

Parkinson's disease is a long term degenerative disorder of the central nervous system. Some of the most prominent symptoms caused by Parkinson's disease can be observed from the motoric abilities in patients hands. The disease often causes tremors, slowness of movement, rigidity and postural instability. At home, these symptoms can be measured with both the angular and linear accelerometers in a modern mobile phone. The progress of the symptoms can be followed with these measurements. When the regularly measured data is combined with medication information, a more comprehensive view to the clinical picture can be achieved. [10]

Many Parkinson's disease patients experience the so called On/Off phenomenon, which is more common during the latter parts of the disease. On/Off phenomenon can be described as a fluctuating severity of the disease. During the On-phases the medication is typically more effective, which means fever and less harsh symptoms. During the Off-phases the medication is typically less effective which can be experienced as more intense symptoms. The changeable symptoms create an even bigger need for tracking and visualizing the overall progress of the disease. [10]

Parkinson's disease patients have also non-motor symptoms especially when the disease worsens. The Sydney Multicenter Study of Parkinson's disease evaluated patients and found out that 84% of them showed cognitive decline and the 48% met the diagnostic criteria for dementia after 15 years of follow-up. Another study found out the Parkinson's disease patients have up to six times the risk of dementia. Parkinson's patients suffer from mild cognitive impairment which occurs as mild memory loss and thinking problems. The term 'memory loss' can be misleading but the symptoms include, but are not limited to, (1) slowed thought processes, (2) difficulties with planning, making decisions, following and taking part in conversations, (3) poor concentration, (4) lack of motivation and (5) short-term memory loss. Especially short-term memory loss and slowed thought processes can be irritating when using modern applications, as they might cause an information overload and there might be lots of quick pop ups. [11, 12]

2.2. STOP

A mobile phone application for measuring tremors, hand movements and medication data has been developed at the center of ubiquitous computing at the university of Oulu. This application is called STOP, and will provide us with the actual data for our project. STOP consists of two main parts: a ball game, which the patient is prompted to play regularly every day and a medication dosage logger, which the patient uses to log medication that has been taken. In the game part STOP measures tremor-like symptoms by utilizing the accelerometers of the patients mobile phone. The accelerometer data gathered during a game session is then reduced to a score, which represents performance during the game. The scores are then stored using the AWARE-framework [13], which is an integral part of the STOP-application.

2.3. Medical Adherence

Adherence to a medical regimen is used to measure the patient's ability to follow their prescribed regimen for a particular medicine or therapy. Knowing patient's adherence to a medical regimen is important in medical research. Adherence variations between patient's can distort the difference between treatment and placebo. [14]

One method to assess adherence to a medication regimen is by looking at the drug level in biologic fluids. The confirmation that the patient has recently taken a dose of drug can be seen in a presence of a drug, or its metabolite, in a biologic fluid. However, multiple patients' serum levels of the drug could be on the same level, but each may have consumed the medication very differently. This means the presence, or absence, of the drug does not equal to compliance or noncompliance.

Biologic markers are easily detected compounds that can be added to medications. Possible usage of a marker would be in a clinical trial where the marker is added to the target drug or placebo. Biologic markers and biologic fluids have the same characteristics. They can provide qualitative data but they have flaws when quantitative measurement of adherence is required.

In clinical trials patients may be monitored that they receive their medications. However, it is possible that the patient can feign swallowing the medication and removing it from the mouth when they are not being monitored anymore. As it is, this method is not practical or even required in most situations.

It is possible to assess adherence to medication regimen by letting the patients explain themselves how they take their medication. This method is generally reviewed as unreliable for accurately assessing adherence. Patients who report poor compliance are generally more correct, whereas patients denying poor adherence may not be. It should be noted that a study by Morisky et al has found 700 studies that used more than 200 variables to assess adherence. One of the problems in patients self-reporting is the number of methods used to interview and retrieve information from patients. Previously mentioned methods include: 1. patient-kept diaries, 2. patient interviews with specific questions regarding the accuracy of medication regimen adherence and 3. standardized, validated, adherence-specific questionnaires.

The interviewer's skill and the construction of the questions can affect the accuracy and validity of patient's answers. Interviews regarding compliance has been referred as interrogations. Patients who joked, laughed and sought suggestions from their physician had the highest rates of compliance. This mean that relationship between patients and the health care professional affect the compliance. The interviewer's wording of the questions can affect how the patients will interpret the question. A question with negative wording (a question which seems to blame the patient), will be answered with bias. Thus, it is difficult to use patient interviews to assess studies because the answers depend on how the questions were constructed and the manner in which they are asked.

One of the more common methods, pill count, is used to determine medication regimen adherence. A pill count is simply counting the returned dosage unit in an appointment and comparing the number of units received in the most recent prescription. Instead of pill count, it is possible to use a device, such as metered-dose inhaler, where the number of inhalations used can be calculated from the change in canister weight. Pill counts are simple and low cost, which is a reason pill count is used frequently in clinical studies and medical practice. However, the accuracy of pill counts can vary widely when estimating the actual adherence with medication regimen.[14]

2.4. User Interface Design and Usability

Acting as the primary point of interaction between the user and the application, it's important to design a user interface that is easy to use for all types of users. The user interface should have good readability. The elements that have an effect on readability are the style and size of the font and a matching contrast between different parts of the interface. The navigation style of the user interface should stay constant throughout the application [15, 16]. User interfaces usability should be carefully thought out and there has been lots of research on usability and on user interface design.

The rapid growth of technology has left elderly people in potentially uncomfortable situation. They may have a limited experience on computers due to economical, physical, sensory and cognitive barriers [17]. Because of the fear of the consequences of using unfamiliar technologies, older people have been relatively passive adopters of new technology [18]. Elderly people have decline in motor abilities and we can equate the decline to Parkinson's disease patients symptoms. Usability of computer interfaces for elderly people has been researched, and as a result there has been developed design guidelines for elderly and people with disabilities [19].

2.4.1. Usability

Every object of use has a level of usability, which can be measured. Usability is the ease of the use and learnability of the object. Usability in software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use. ISO9241-11 standard defines usability as "the extent to which a product can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use" [20].

Jakob Nielsen defines usability with five quality components: learnability, efficiency, memorability, errors and satisfaction. Learnability means the ease of use for users to accomplish basic tasks the first time they encounter the design. Efficiency means how quickly user can perform tasks when they have learned the design. When the user return to design after a period of not using it, memorability tells how easily they can re-establish proficiency. Errors includes usual error of users, the severity of error and the error handling, as in recovering from the errors. Satisfaction is the pleasure of using the design. Besides these five quality components, there are also other important quality attributes. For example, one is utility - does it do what users need? [21]

Usability has different methods to measure usability, for example a needs analysis. The elegance and clarity of interaction between a user and a software or a website is studied in human-computer interaction and computer science usability studies. Usability considers user satisfaction and utility as quality components, and aims to improve user experience through iterative design. Usability measurement can be design reviews or empirical measurements. Design review is a usability-inspection method, where one reviewer examines a design to identify usability problems. Design reviews can be conducted at all stages in the design cycle as long as there is a prototype of the system with enough level of detail. Real users do not use the system in design reviews, the reviews are based on inspection. [21, 22]

Different types of design reviews are for example: heuristic evaluation, standalone design critique and expert review. In heuristic evaluation, design is evaluated for compliance with different heuristics, such as Nielsen's usability heuristics. In standalone design critique, in-progress design is being analyzed. Analysis can be performed as a group conversation and it determines whether the design meets objectives and provides a good experience. In expert reviews UX experts inspects the system. Expert reviews are close to heuristic evaluations and it is okay to think of an expert review as a more general version of heuristic evaluation. [22]

Empirical measuring tests the system on real users using behavioural measurements. This includes testing the system for learnability and usability. Quantitative usability specifications such as time and errors to complete tasks and number of users to test are used in this stage of measurement. Reviewing or demonstrating the system before user testing can affect the results of the tests. The users should be as close to end-users as possible or the results are not as correct as they should be. According to Nielsen, best results come from testing with three to five users, and running as many small tests as possible. According to Nielsen, elaborate usability tests are a waste of resources. [23] Tom Landauer and Nielsen have showed that the usability problems found in a usability tests with n users is:

$$N(1-(1-L)^n), \tag{1}$$

where N is the total amount of usability problems and L is the proportion of usability problems discovered while testing a single user. The value of the L is typically 31%. Plotting of the curve from the (1), when L=31%, can be seen from the figure 1.

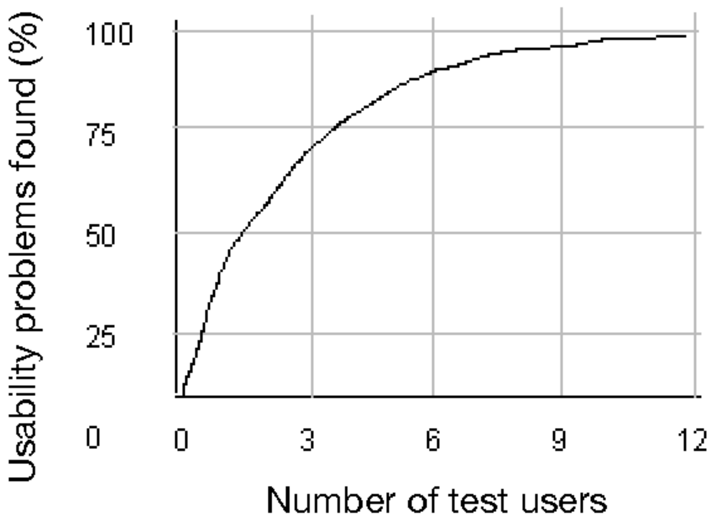

Figure 1. Proportion of usability problems found.

Other empirical measuring methods are paired-user testing, visual walkthrough and informal walkthrough. In paired-user testing two users perform predetermined tests while talking about them. It has been found that thinking aloud is more natural when users can ponder to their peers. It is recommended that users' skillsets are close to each others as otherwise results might not be as accurate. Paired-user testing is one of the most popular user testing methods. In informal walkthrough users test the system without and separated instructions or predetermined tests. The tests should still be designed with care because informal walkthrough requires specific action plan and definition of the goals to be effective and successful. This methods effectiveness depends on user's information, skill level and experiences. Visual walkthrough collects data from the user's perspective. It tells how the user interprets the system and its functionality. In the test settings users are asked to inspect the product, describing what they see and to discuss the meaning of the products functionalities. The goal in this is to understand how the user examines the product and where the user pays attentions. [24]

2.4.2. Gestalt Laws

Usability, measuring it, and its measuring methods share a common problem—usability is relative. People experience usability in different ways. User's background influences their way of using software. Tech-savvy users might use programs with intuition differently than other users. Even the background of skilled users influences the way they use software. For example: education, experiences, age and sex might influence the decision making while using an application. Especially in modern software, there are lots of different user interfaces. Majority of people might find command line interfaces as difficult to approach and non-intuitive, while programmers might prefer to use command line interfaces instead of graphical user interfaces.

Gestalt laws are important in the user interface design, as they might improve the products usability. There are such laws as: law of completeness, familiarity, proximity, similarity, continuity, closure, good shape, symmetry, common fate and area. Gestalt laws aim to show the principles of how our brains combines entities from the details of the observations. Different Gestalt laws makes designing user interfaces possible in a way, where the user can visualize them in desired way. According to law of completeness, the "wholeness" is outlined first, even when there are other visible details. By the law of familiarity, if the shapes are familiar, people tend to figure them out first from the other figures. In other words, the objects to be examined are more likely to form groups if they have known or meaningful shaped upon the maker.

According to law of similarity objects with similar shapes and colours are associated to the same group. Similarity can be grouped to three factors: shape, size and colour. The size of the objects has been found to be more significant detail than shape of the objects. Also, colour and its strength creates a strong similarity between objects. Even though the strength of the colour is only a part of itself, it can affect as independent motive. In some cases, colour's strength can act as a colour itself and that way it is usually even more stronger element. Grouping objects is also possible with shapes. By the law of proximity closely related patterns are perceived to be cohesive. Proximity can be even more stronger power mean compared to similarity. The law of proximity and law of similarity can be combined, creating the most powerful impressions of grouping. The closer the objects are to themselves, the more likely we will associate them as a group. However, there are multiple types of proximity - being close, touching and overlapping. When objects touch, they are still separate objects but they are associated as a group. This proximity is stronger than closeness, thus forming more noticeable ties within the group. The strongest proximity is overlapping, where two or more objects overlap. Overlapping can create illusion of depth. Especially overlapping with different coloured objects are fastest to notice.

According to law of good shape, patterns are understood as simple and symmetrical. In other words, law of good shapes shows us how a person seeks patterns as simple, good and regular as possible. Thus, person tends to detect simpler pattern than they appear after a more detailed examination. Law of good shape is especially important in mobile design where objects should be displayed as simple as possible. Similar views lead to us associating similar designs to common mobile designs like front pages and calendars. Law of symmetry tells us we are more likely to combine parts as a one pattern if the pattern is symmetrical, rather than seeing the parts as individual patterns.

By law of common fate, we associate objects as a group if they seem to be moving to the same way at the same speed. The law presents that the objects moving the same way are more likely to belong to same group. This means that individual movements do not separate from the group. According to law of closure, closed, or almost closed line creates a figure. Law of closure is one of the most used Gestalt laws. If visual objects seem to close an area inside them, people tend to associate the area as its own group. According to law of continuity consistent line is perceived as a pattern. In other words, the law continuity describes how elements of the same objects can express some kind of continuity. If the lines intersect, the viewer divides the whole into clearly continuous parts.

One should take into account of possible groups, where user interface elements can be divided into, when designing user interfaces. The elements should be arranged into logical groups in order to make use of user interface easy and intuitive. User interface designers should remember that Gestalt laws are only guidelines after all. The laws can also be used as in contrast. Placing groups far away from each other creates stronger association of two groups.[25]

2.4.3. Evaluation and Testing

In functional testing, also called black-box testing, tester do not see inside the component. The component itself is rather a black box. Testing is therefore based on an externally detectable activity. Black-box testing usually tests the value range of input variables. In other words, the main intent is to break the program with incorrect boundary values. [26]

In usability testing any kind of usability problems are tried to be found from the prototypes. The idea of usability testing is to test the program as early as possible. If the program would be, for example, in beta-version, no major changes might be made because there would be no time to make the changes or they would cost a lot. End users are ideally included in usability testing to provide proper information for planning decisions. The biggest difference in black-box testing and in usability testing is that in black-box testing one wants to break the program. In usability testing information about the usability is gathered.[27]

Heuristic evaluation is usability inspection. Usability inspection does not need user testing at all, instead evaluators examine the program and judge its compliance with heuristics. Evaluators are usability experts and the examining relies on their experience. Heuristic evaluation does not need but one expert, thus it being affordable and fast option for evaluation. The evaluation is intuitive - it does not need lots of planning and it can be performed in the early stages of the development. The number and severity of design error discovered by user testing can be reduced by using heuristic evaluation before user testing.[28]

The evaluation of usability can be divided into the expert reviews and empirical user tests. Heuristic evaluation is a one example of expert reviews because the review is done by an expert without actual end users. The experts review the user interface from the perspective of usability design heuristics. There are many sets of different heuristics, for example very commonly used Jakob Nielsen's heuristics and Gerhardt-Powals' heuristics.

Empirical user tests are experimental and the end users are integrated to the review. Usability testing is a one example of empirical user tests. Choosing the method of evaluation depends on various phenomena. One should think about one's priorities, whether it be efficiency of use or learnability of use. Also, the goal of the evaluation effects to choosing the method of evaluation - does one want to develop better product or does one want to test a new idea. Both empirical user tests and heuristic expert reviews produce different kind of results, thus it would be considerable to combine both methods.[24]

2.4.4. User Interface Technology

On top of providing an interface for the user, many dynamic websites are constructed to use a data-provider service, often called a backend. Backend is a server-side software, which is explained later in this document in section 2.5. REST-architectures (Representational state transfer -architectures). Dynamic websites consume the data provided by the service and may also post data back to the service. The communication between a backend and the user interface utilizes HTTP (HyperText Transfer Protocol). The delivered entities or sets of entities are chosen according to requests made by the frontend. A single web application may utilize multiple services to implement its functionality. For example, the application could ask data from one service and then pass this data to another service for analysis. The analysis could then be presented in the webpage dynamically. This allows individual development for interface and service control logic. [29, 30]

Web applications' UI's consists of HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) and ECMAScript. HTML is document description language and HTML elements are blocks used to build HTML websites. HTML was used to handle all the aspects of the web pages or documents, as all the web was static documents. Nowadays HTML is one of the cornerstones of the World Wide Web, alongside with CSS and JavaScript. [31]

With CSS one can separate the style of the web document from the content. In other words, CSS is used to style web documents. Cascading the styling rules allows the styles to be brought in from several sources and they are cascaded in a user agent as one. It is also possible to present the same markup page in different styles for different rendering methods.[31]

ECMAScript is a scripting-language specification, created to standardize JavaScript. JavaScript is the best known implementation of ECMAScript but other known implementations exists, such as JScript. ECMAScript is interpreted in runtime, allowing the modification of the code, for example, by adding function or modifying variables, in runtime. ECMAScript is used for client-side scripting on World Wide Web.

Programming standalone application, the UI is just a program component, implemented with UI toolkit with the same language as the components. Such widget toolkits, known as GUI frameworks, are such as Swing for Java [32], GTK+ for C [33], Qt for C++ [34] and Cocoa on Objective-C. [35, 31]

2.4.5. User Interface Design

User interfaces in the field of information technology have come from batch interfaces to command line interfaces to graphical user interfaces. Later on when talking about user interfaces we mean graphical user interfaces. Great interfaces have common qualities and characteristics, which include: clarity, concision, familiarity, responsiveness, consistency, aesthetics, efficiency and forgiveness.

Clarity means that everything in interface is clear, throughout the process from the language to flow and to visual elements. With concision we try to avoid the interface bloat where there is too much stuff on the screen at the same time. Interface bloat causes information overload and finding what you are looking for gets difficult. If someone uses the interface for the first time, should one still be able to use the interface because of the familiar elements. A good interface gives good feedback to user. Whether it be what is happening or what is happening at the moment or if the user's input was processed successfully. Attractive interface makes using the application more enjoyable. Even if the interface's aesthetics do not improve the functionality or the ease of use, it improves the user experience. A good interface is efficient and it ought to have good shortcuts and intuitive design to be able to be used efficiently. A great interface babysits the user. Rather than punishing user for it's mistakes, instead it provides the means to remedy them.

When designing user interfaces, one should gather the functionality requirements first. User stories are a great way to obtain information on different use cases, and possibilities, limitations and goals related to interface. Personas are fictional characters which represents different type of users that use the software. Personas describes different types of personalities and their different needs, experiences and behaviour. Personas help us to visualize possible scenarios and thus helping us to plan the functionalities and interfaces flow. Scenarios are created from the collected material, like user research, and from the user stories.

The design itself is started with information architecture - with the systems structure, navigation and the flow of the system. Information architecture describes structures of the data and it aims to parse the content, elements and also search and other possible features and their relations to usable structure. With information architecture we can gather information, like priorities, and create the user interface to be easy to use. We combine information contents and technology into an entity understood by users. User is able to understand in which part he is in at any time, how he got there and how he can access other parts of the service, if the information architecture is successfully done.

With the information gathered from information architecture one can develop wireframe prototypes. Prototypes can be either paper prototypes or simple interactive screens. The prototypes are stripped of fancy looking objects and elements. The wireframe is accompanied by explanations like why the different controls exists and what they do. The prototype also describes both behavior of dynamic objects and how dynamic websites refreshes. Prototypes work as a documentation and it allows more specific design and it passes information on design solutions.

The actual content for users is presented as text, pictures, videos, animations and audio. Other elements in the system exists to help the user find the right content. The user experience of using the product is largely due to finding the right information or content and how well the content fits into the user's goal. With visual design, we can achieve better efficiency of product use and ease the use for new users. Working speed can be improved by twenty to forty percent with a good design of the screens and development of the visual features. The interface's layout of elements should ensure a good communication between the software and the user. The layout of the components should be designed with user's goals and tasks in mind. [28]

2.4.6. User Interface Design for Elderly and People with Parkinson's

As mentioned in part 2.1 people with Parkinson's disease are restricted in many ways and therefore the design of the user interface should be carefully thought. 95% of Parkinson's disease diagnoses are diagnosed on people above age 60[36]. Meaning most of the guidelines for designing user interfaces for old people applies here. For people with Parkinson's distinguishing between low contrast elements can be hard as the visual system might be affected during the disease. Therefore, using high contrast elements in the design is recommended. This also follows the guidelines for designing to old people. [19]

A common symptom for people with Parkinson's can be short-term memory loss. Therefore, overloading the view with too much information can be seen as overwhelming. In addition to avoiding too much information, complex and nested user interfaces should be avoided as one might forget what they were doing or where in the user interface they are. These suggestions also follow the guidelines for designing for the elderly. The off phase especially can cause slowness of movements and thus time-dependent interactions should be avoided. For example, a pop-up chat which automatically closes after a few second can be seen as frustrating.

The following guidelines are meant for touch interaction, such devices can be smartphones or tablets for example. The size of elements should be larger than for typical users. Suggested size for tap elements was evaluated from usability experiments and a good value of 14 mm side was found with a 97% accuracy. Since the slowness of movements, the swipe gestures activation speed should be removed in some cases. The drag gesture falls into the same category with the swipe gesture, the slowness of movements can make these gestures frustrating and hard to execute and should thus be used carefully.

There are twelve design guidelines for people with Parkinson's disease. These design guidelines are split into two groups: touch interaction and information display. Touch interaction DG (design guidelines) are following: DG1: Use tap targets with 14 mm of side. DG2: Use the swipe gesture, preferably without activation speed. DG3: Employ controls that use multiple-taps. DG4: Use drag gesture with parsimony. DG5: Prefer multiple-tap over drag. DG6: Adapt interfaces to the momentary characteristics of the user. Information display are: DG7. Use high contrast coloured elements. DG8: Select the information to display carefully. DG9: Provide clear information of current location at all times. DG10: Avoid time-dependant controls. DG11: Prefer multiple modalities over a single interaction medium. DG12: Consider smartphone design guidelines for older adults.

The design guidelines were developed by reflecting on the studies where participants with Parkinson's disease performed a set of tests, including taps, multiple taps, sweeps and drags. The study showed that when tapping buttons, the size of the target influences the accuracy of the tap and the spacing to surrounding elements does not. Button with 14 mm side had the best mean accuracy. Though participants were able to perform the swipes, and there was no significant correlation between the target height and spacing between target and surrounding elements for the swipe gesture, participants swiped at very different speeds. Supported swipe speed should be 24 mm/s to include around 95% of the study's participants. Participant were able to perform successive taps with no significant reduction in speed up to ten taps and they were also able to drag objects with precision over a sale of at least ten elements. However, participants were slow to reach their goal by dragging. In fact, some participants manifested some frustration while performing the dragging tests.

Because the Parkinson's disease can impact vision and cause a short-term memory loss, should the user interface take it into account by using high contrast elements and reducing the amount of information displayed. Also because of the short-term memory loss and slowness of thought, the interactions with smartphones are not fast. The current location reminds users of what they want to achieve and quickly informs them in case they select the wrong target. Having a time-dependent controls can be difficult and stressful because of the movement speed reduction. This effect increases during the Off phase. It is good to have multiple modalities for the same control because Parkinson's disease can impact both vision and speech. [19]

2.5. Time Series

Studying a patient's well-being over periods of time is crucial to know whether the treatment is working. This is even more true for people suffering from long term illnesses where data accumulating over the years can be an overwhelming amount. It is important to view the data over time and see whether the data changes overtime by itself or in relationship with other features.

Time series consist of trends and cycles which can be on different levels [6]. These factors make time series work in a systematic way. For example when the temperature cools down in the evening, a daily cycle can be seen. The temperature also decreases in the winter, here a yearly cycle can be seen. Time series can also have random cycles. For example when a high pressure area arrives the temperature rises.

Predicting the future is a very common problem for many fields such as finance, signal processing and astronomy. The problem can be simplified to predicting the next datapoint in a time series. Typically the best prediction for the future is the newest datapoint. For example, the best prediction for tomorrow's weather is today's weather.[19]

2.5.1. Visualizing Time Series

Time series rely on the temporal domain being on the most important feature of the data and thus it is most often used as part of the visualization. For cyclic events such as signals where the varying frequency of the time series is seen as the important feature a frequency domain can be used for the visualization.

Time series can be visualized in three different ways in the temporal domain: linear, cyclic or a branching tree. Linear representation can be seen as the same as we understand time, it consist of ordered time points, i.e. there is a past, a now and a future. Cyclic representation consist of a limited time interval with recurring cyclic events. For example if one was interested in determining which day created the most traffic, a sensible solution would be to plot the data in a Monday to Sunday axis time interval where possible fluctuations could easily be distinguished. Branching tree is modeled as graphs. The vertices are moments in time and the edges represent the direction of time. Branching trees are used for predictions and alternative situations. [37]

2.6. REST-architecture

REST or representational state transfer is a web-architecture originally developed by Roy Fielding in 2000. REST presents more of a guideline for developing the external interfaces of a web service rather than providing an exact specification. A large portion of the interfaces in the modern world wide web, which provide a view to a resource, follow the REST architecture. Some of the reasons for using the REST architecture are its good scalability and clarity, which results in easier and faster development. A correctly deployed REST interface does not require as many HTTP requests to achieve its goals and it also lowers the latency of the network by utilizing proxies. [20-22]

Examples of existing REST interfaces are abundant. For example Twitter [38], Wolfram Alpha [39], the city of Oulu [40] and Youtube [41] amongst many others provide REST interfaces to implement their services. Some of the interfaces and endpoints are entirely open, which means anybody can utilize the endpoints as they please. Other interfaces implement some sort of access control, which requires registration. Some of the key characteristics of the REST architecture are client-server architecture, statelessness, cacheability, layered system, uniform interface and the ability to provide code on demand. Uniform interface is an umbrella concept for self-descriptive messages, hypermedia as the engine of application state, resource manipulation through representations and resource identification in requests. [29, 42]

REST-services only provide an interface to manipulating a set of resources. This means, that they are not meant to be used by humans, but rather consumed by a dedicated interface software. In the web world, this is called a frontend, and usually developed for a web browser. Frontend is what we introduced in section 2.2. In figure 2 is described to high-level architecture, which consists of a web client, RESTful service and optionally a database. Client and server communicate via HTTP methods and the server is connected to the database via varying protocols, depending on the case. [29]

Typically REST-interfaces or APIs (application programming interfaces) implement the CRUD-methods, which is an abbreviation for create, read, update and delete. The CRUD-methods correspond to their relative HTTP-method counterparts as POST, GET, PUT and DELETE. POST-method is used to create a new entity in a set of entities. PUT overwrites an entity, identified by some of its attributes. DELETE deletes an existing entity or a set of entities. GET retrieves an entity or a set of entities. The payload, which is delivered in the HTTP body may be of arbitrary format, although JSON (JavaScript object notation) is very common data type. [30]

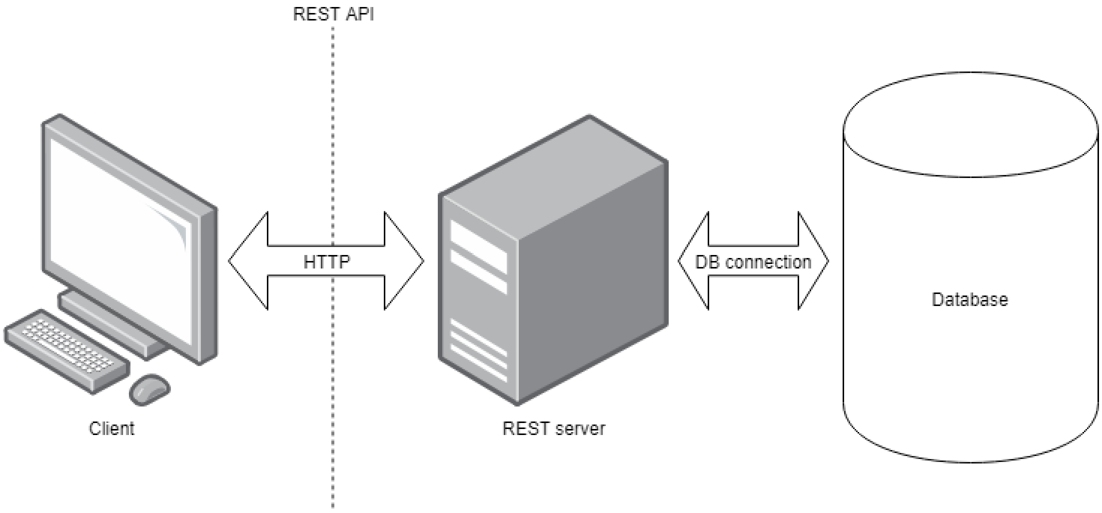

Figure 2. Typical components in a client-server architecture in web environment.

An application programming interface that is developed with the REST-guidelines adheres to the rule, in which resources or sets of resources are identified by a unique address. The usage of uniform resource identifier or URI is defined in RFC 3986 [43]. For example, an address ‘https://medidash.com/api/v0/user/1337/scores’ identifies the scores of a user with a unique ID number 1337 in the MediDash API. Figure 3 explains how our application should retrieve data for visualization from the database, through the API. MediDash is a typical example of an application utilizing the client-server architecture in the World Wide Web.

2.7. Python Libraries

2.7.1. Dash

Dash framework is a open source library for Python. Dash has been built on top of Flask [44], Plotly.js [45] and React.js [46]. The framework itself is for building web applications and Dash provides tools for easy data visualization. Dash abstracts away the technologies and protocols needed for building interactive web applications which are cross-platform and mobile ready. [47] Dash applications are web servers. They are communicating with JSON packets over HTTP requests and run on Flask. Frontend components are rendered using React.js. The instance of Flask with it’s all configurable properties is accessible to Dash app developers. Dash apps can also be used with Flask plugins as well.

Dash provides toolset to package React components, that way React components can be easily used in Dash. The toolset uses dynamic programming to automatically generate standard Python classes from React propTypes. This results in Python classes which are user friendly. They come with automatic argument validation, docstrings and more. Dash component libraries are imported separately from the core Dash library. This means you can write or port a React.js component with React-to-Dash toolchain. Because Dash application saves states to the front-end, Dash apps can be used in a

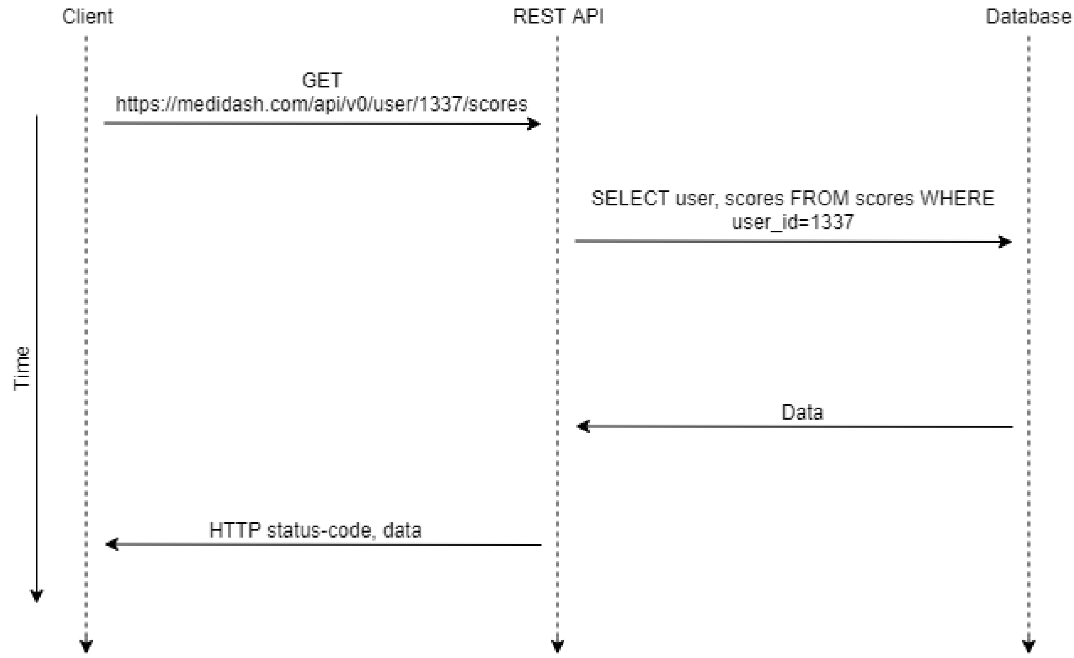

Figure 3. Simplified example of communication between the client, REST API and a database.

multitenant setting. Multiple users can have independent sessions while interacting with Dash app at the same time. Dash's core library has no CSS and default styles so Dash apps styles have possibilities to be customized.[48]

The Javascript library Dash uses to render its dynamic components is called React.js. React.js is a JavaScript library, maintained by Facebook and community of developers, for building user interfaces. React.js is suitable for development of single-page and mobile applications. React.js supports additional libraries, like routing and interaction with an API. [46]

React uses a feature called "Virtual Document Object Model", or "virtual DOM" (Document Object Model), where a representation of a UI is kept in memory and synced with the 'real' DOM by a library. Virtual DOM allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change. Virtual DOM has been criticized, as it has high RAM-memory requirements. [49, 46]

2.7.2. Flask

Dash applications are web servers running on Flask and they communicate with JSON packets over HTTP requests. Flask does not require particular libraries or tools, except for some basic standard libraries, making it a micro web framework and it is written in Python. [44] Flask is based on the werkzeug utility library and Jinja2 templating language for Python. [44]

Flask has no database abstraction layer or any other components. However, Flask has extension support and in that way application features can be added as if they were implemented in Flask itself. Extensions can be for example open authentication technologies and form validation [50]. Flask framework is used in Flask's own community page. [44, 51, 52]

2.7.3. Plotly.js

Plotly.js is a open-sourced JavaScript library, ideally suited for graphs and charts using applications. Plotly.js uses D3.js and WebGL for graphics rendering and it is an "all-in-one-bundle", meaning everything is included when installing plotly.js [53]. Plotly.js is versatile tool for creating charts as it supports lots of different data visualization resources. For example Plotly.js can create scatter plots, different kind of charts, like pie and bar charts, histograms, 2D density plots, heatmaps, parallel coordinates plots, financial charts, maps, 3D charts and many more. [45]

Plotly.js is comparable to MATLAB and Python's matplotlib but Plotly is more suitable for web because it is written in JavaScript. WebGL allows interactive rendering of millions of x-y points and D3.js is for vector-quality image exports. Plotly.js has image testing framework, making it the most stable JavaScript charting library available. Plotly's user documentation is hosted on GitHub pages and is open-source. Plotly has no dependencies and it has no jQuery, improving it's performance and browser compatibility. Plotly is based on an open-source JSON schema. [54]

2.7.4. Numpy

NumPy is a Python library, which extends Python with large, multi-dimensional arrays and matrices, and high-level mathematical functions, such as linear algebra, Fourier transforms and random number capabilities. NumPy can also be used as a multi-dimensional container for generic data. NumPy has been widely adopted in academia, national laboratories and from gaming to space exploration industry. Python's for loops are often a bottleneck when computing big arrays but with skillfully applied NumPy most of the computation time is spent on vectorized array operations. It is possible to achieve even further speed by using optimizing compilers, such as Cython. [55]

2.7.5. Pandas

Pandas is a open source data analysis and manipulation library for Python. Pandas provides data analysis tools, data structures and operations for manipulating numerical tables and time series. Pandas aim to be the fundamental high-level building block for doing real-time data analysis in Python. The library's critical code path is written with Cython or C, making it highly optimized for performance. Pandas' main features are such as (1) easy handling of missing data, (2) data structure column insertion and deletion, (3) intuitive merging and joining data sets, and (5) time series-specific functionality, like data range generation and frequency conversion.[56, 57]

3. IMPLEMENTATION

The implementation of this project loosely follows RESTful principles. As a backend we have a Flask based Dash application, which handles data manipulation, fetching, transfer and even authentication. Flask and therefore also Dash utilizes both Jinja2 as a templating engine. Dash also uses ReactJS library to implement its interactive plot elements. The general flow of information goes so that when the user navigates to an URL eg. ‘https://www.medidash.com’, they are redirected to a login page. In the login page, the user types in his account credentials. During the authentication, the backend allocates each user and their session a session token to keep the user logged in. After the authentication, the user will be redirected to the actual application in URLs medidash.com/regular or medidash.com/advanced. From these views the user is able to access either a simplified view (/regular) or a detailed view, which provides more information. The information shown consists of graphs, numerical data tables and timelines or calendars. All of the informatics are generated from the users personal data. The authentication process provides the device ID to the backend. This allows the backend to retrieve the correct data from the Aware-framework database. STOP application sends the user data to the Aware-framework database. User can give access of his or her data to another user, for example a doctor or a relative. The dashboard’s flow chart can be seen from figure 4.

AWARE is an open-source Android framework which captures hardware-, software- and human-based data. AWARE is dedicated to instrument, infer, log and share mobile context information. AWARE provides an SQL database and the required connectivity to build sensor applications on top of. AWARE tries to help users of the framework by transforming the data into information user can understand.

AWARE lets users record their data from the phone without any programming knowledge. Users do not need to worry about the privacy as AWARE does not log personal information. AWARE enables easy mobile related studies. Researchers are able log data by installing AWARE to the participant’s phone. With AWARE dashboard researchers are able to request the participant’s data. Everybody can also create plugins for the AWARE. The STOP application this project is based upon utilizes AWARE in implementing its functionality.

The operating environment and use cases revolve around two very specific topics. One use case is a patient that views his or her own data. The application allows a patient to keep track of the symptoms and taken medication. The second use case is a specialist, a doctor that views the patients data to draw suitable conclusions from. The same data should be accessible to both parties, because it is not ethically good to spread patient data to another user, which the patient cannot see themselves. This means that not all the intricacies should be shown as a default, but rather be expanded upon on demand. This is why there are two views, which have different levels of detail. One is aimed towards the average user and the other is aimed towards a specialist looking for a more in-depth analysis.

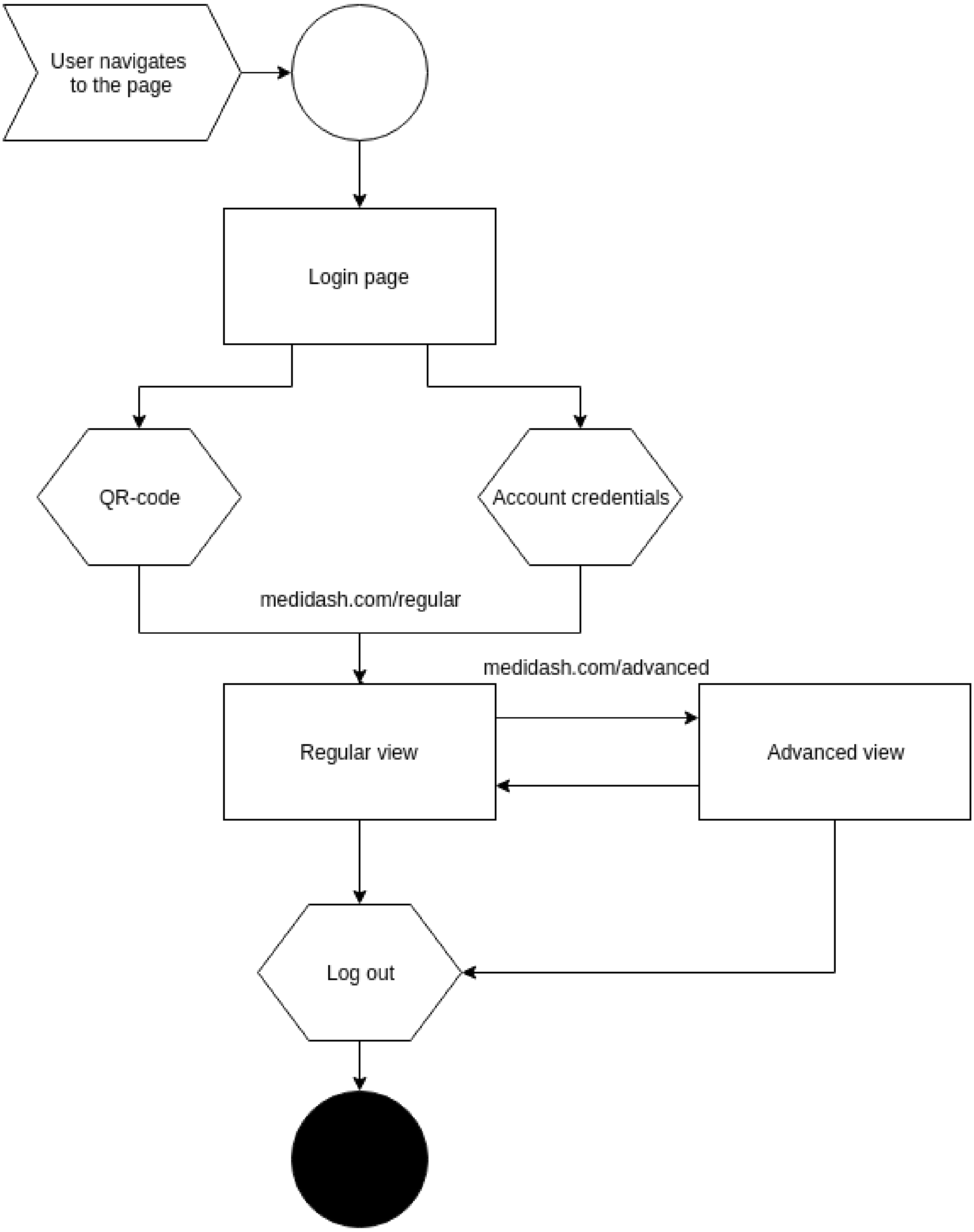

Figure 4. Flowchart of the users navigation in the application.

3.1. Model-View-Controller

Python Dash applications use the model-view-controller (MVC) software architecture pattern, which is visualized in figure 5. MVC-pattern is commonly used to develop user interfaces. Our applications can be divided into three components - model, view and controller. Model is the central component in the pattern, independent from the user interface. It manages the data and the logic. View represents information, like charts and tables, this is what the user sees. User uses the controller, that takes inputs and convert them to commands for the model.

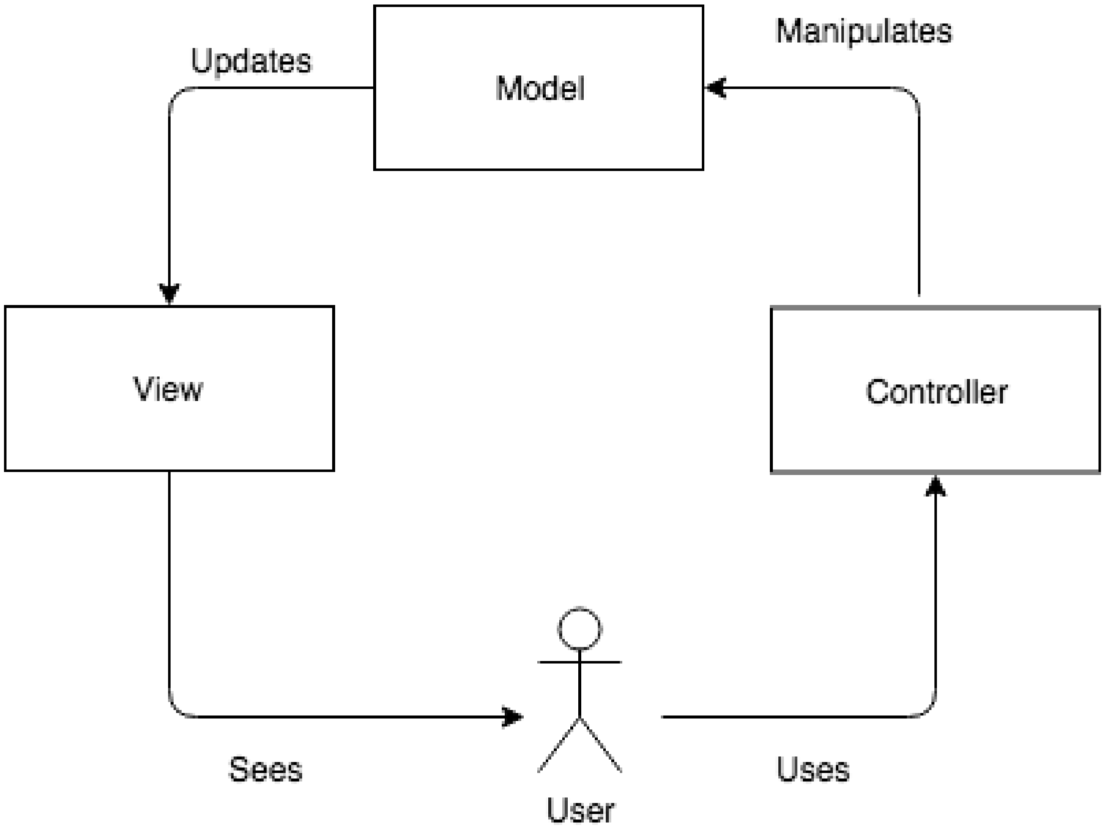

Figure 5. Model-view controller

3.2. User Interface

In this section we will discuss our implementation of the user interface and the graphs in it. UI plays a crucial part in the implementation as it is the final product the user sees. As the main interest for the user is seeing their data they have gathered we will look at different visualizations in section 3.3. Especially important when designing an UI is to consider your user group and the constraints given by them. We will look at these special requirements in section 3.3.4.

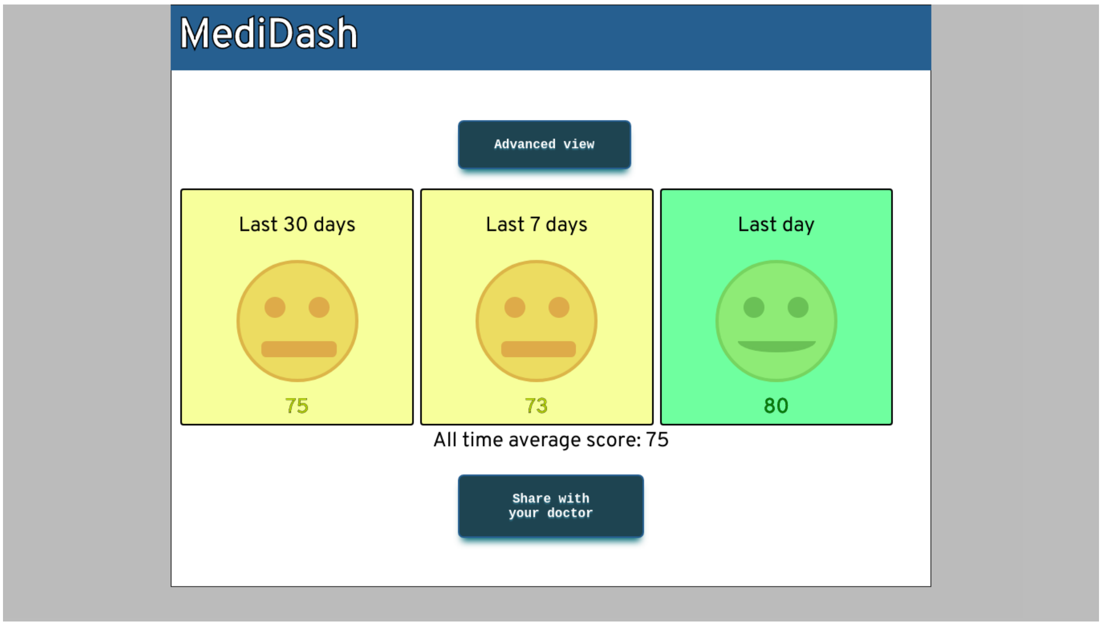

Figure 6. User interface of the regular view

3.3. Data Visualization

As our data consists of different values between different timepoints our data is to be considered as a timeseries. A typical time series visualization is a line plot with time being on the x-axis and the values being on the y-axis. This type of representation gives a look at the whole data within one glance. Given many data points over a long time the plot can start to look clustered and hard to follow especially if the standard deviation is great in a short window. To solve this problem aggregations by a windowing function can be made. For example aggregating all the values within a day and calculating their mean declusters the data and makes it smoother by evening the outliers. In our case the word "outliers" might not be good, a better word to describe these unusual values would be "inconsistencies" as these unusual values might give high indications that something is wrong.

3.3.1. Visualizing the Game Scores

The most common way to represent a time series is with time being on the x-axis and the values measured at different time intervals on the y-axis. In our implementation on the advanced view (see figure 7 we have given such visualization with score being on the first y-axis and medication taken on the second y-axis. The ability to modify the viewed time span is given in a typical calendar form which is made easy to use and recognizable from other sites that provide the same ability. A linear line to help distinguish with the trend was also fit with the ordinary least squares method. More details of this are given in the next subsection. By default the application shows all of the data recorded from when the first game was played till when the last game was played. This allows to look at the long term trend. To restrict the time interval the user can use calendar located on top of the plot to choose proper dates according to

Figure 7. User interface of the advanced view

their needs. To help the search for a trend and filtering out outliers a rolling window function was added. With this the user can group the scores by a day or a week. For example grouping by a day averages all the scores from a day and presents that value.

Although following the game scores is important the same could be said for the amount of medication taken on a particular day. In the advanced view we have added a bar plot in addition to the line plot from game scores. This allows the user to see both measurements with a glance. All of the above modifications done on the game scores plot will also be applied to the medication taken bar plots making it easy to follow both.

Regular view was made as easy as possible to follow and understand such that even a person suffering from parkinson's would be able to use it. See figure 6 for a screenshot of the regular view. Only the aggregated values of the past 30 and 7 days and last day are given. The true score values of the game might seem arbitrary so a comparison is made with the past scores to give the impression whether the given aggregated value is good, bad or neutral. This is visualized with colors of green, red and yellow respectively as well as smileys.

3.3.2. Calculating the Trend Line

Given fluxuations in the graph with ups and downs it can be rather hard seeing the overall trend. To help the eye catch the overall trend we would like to fit a line stating the direction and the magnitude of the trend. This gives us a linear equation but since the number of samples is much higher than the number of variables the problem has to solved by approximation. The problem can be mathematically described as when approximating by least squares

$$\min_{\beta \in \mathbb{R}^2} \|X\beta - y\|^2, \tag{2}$$

where X is a $n \times 2$ model matrix with n being the number of samples selected and holds the timestamps of the game score values, β is a 2×1 vector that contains the estimated intercept term β_0 and slope coefficient β_1, y is a $n \times 1$ vector that holds the functions values given β, in this case, the game scores.

The problem is to estimate the values of vector β such that we follow (2). Then we can estimate the trend with the coefficient for the slope β_1. Solving this unconstrained convex optimization problem is quite trivial. Simply taking the gradient of (2) and setting it to zero we find the critical points for β. Adding Xy on both sides and multiplying both sides by the inverse of $X^T X$ from left side we get the following:

$$\hat{\beta} = (X^T X)^{-1} X^T y, \tag{3}$$

Typically in literature the ordinary least squares problem is represented in terms of the estimated parameters $\hat{\beta}$, the model matrix X and the response variable y. Equation (3) gives us the solution for the estimated regressor values of the least squares problem. From this we can easily calculate the estimated response values $\hat{Y}$ with

$$\hat{Y} = X\hat{\beta} \tag{4}$$

Although (4) gives us the solution for our least squares problem issues can arise with inversing the term (X^TX). Numerical errors with inverting large matrices can occur and thus often QR-factorization is used. This also speeds up calculations.

$$A = QR, \tag{5}$$

where Q is a $n \times n$ orthogonal matrix and R is a $n \times p$ upper triangular matrix.

Now transforming X as in (5) and placing it in (3) and using properties of transpose and that Q is orthogonal i.e. $Q^TQ = I$. We achieve the following

$$\hat{\beta} = R^{-1}Q^Ty \tag{6}$$

estimator for β.

We created a model matrix X with the person's timestamps of their game scores and response variable y with their respective scores from the games. Using Numpy's matrix operations and functions we can accomplish fast calculation times even for large matrices guaranteeing an interactive experience . Using (5) and calculating the matrices Q and R for X and then calculating $\hat{\beta}$ from (6) we manage to get the intercept and slope values which we can then use to plot a line with (4). Figure 7 shows the trend line as a green line.

3.3.3. Other Plots

In the regular view it is easy to see your game scores for the last 30 days with a glance. We wanted to achieve the same for medications taken. Although the same principle of a regular time series line plot could have been used it fails to capture some important information. From a regular time plot seeing which days you took less medication is hard and unintuitive. Especially when interested in the current month a typical calendar representation works well for the purpose. The calendar was created by using a heatmap divided into a grid where each slot represents a day. Given the heatmap's functionality it is easy to determine which days the user forgot take their medication. The heatmap can be seen in figure 7.

A datatable was also implemented in advanced view in order to view some key statistics. The datatable includes aggregated values from all time, selected values and last 30 days. The aggregated statistics are mean, count, standard deviation, 10th and 90th percentile. These allow the user to dig deeper into the raw values and for example compare only the 90th percentile values which ignore most of the outliers. This can be viewed in figure 7.

3.3.4. Realized Requirements

As mentioned in section 2.4.5, Parkinson's disease patients are restricted in various ways and therefore the design of the user interface should be carefully thought. As most of the Parkinson's disease patients are above the age 50, the users interface should be designed with these limitations in mind. One of the more important aspects for the design of the user interface was to avoid information overload, as a common symptom for Parkinson's disease is short-term memory loss. Other major guideline for the user interface was to include high contrast elements because Parkinson's disease patients might struggle to distinguish low contrast elements from each other.

To avoid information overload, the user interface's 'regular view' is stripped to bare essentials. The main element in the 'regular view' is the three smiley faces with the score calculated by the dashboard. Each smiley face represent a score—from the last 30 days, from the last 7 days and from the previous days. The smiley faces have three colours—red for bad, orange for something in between, and green for good. Red is associated as attention grabbing and provocative. Yellow is not as aggressive as red and green evokes the feeling of safety. These colours are used in similar manner with traffic lights, and together they act as a good indicators on how is your score. Besides the score, dashboard's user interface has two buttons.

The buttons are deep blue coloured and they have high contrast to white background. When hovering the mouse over the button will highlight with a more brighter blue. This indicates that the button is selected. The first button provides the way to share your information with the doctor. The second button directs to advanced view where are more informatic data which might interest the doctors.

Advanced view does not implement the same design guidelines as the regular view because the Parkinson's disease patients are not the target group for the view. However, the button redirecting back to regular view is in the same place and same style as a button redirecting to advanced view. This way the button creates some familiarity. If someone proceeded to advanced view as an accident, the button would be a familiar landmark and an easy way back to regular view.

Our original plan was to use a similar QR-code login system as is used in whatsapp web client. In the whatsapp web client the user is presented with a unique QR-code [58]. The code is then scanned with a dedicated QR-code reading application. This sends an HTTP request to the server, which then connects the unique QR-code with the authenticated phone. Thus, the user is logged in securely. The implementation of such a system proved to be very problematic because of the technologies we had already invested in. Python Dash does not easily support the required websockets and session controls required to login an user with a mobile phone based two factor authentication. So we decided to simplify our implementation and leave the very advanced login methods to be worked on in the future. With the time spared we were able to completely rework the interface.

3.4. Deployment Setup

As Medidash is developed on Python, it makes sense to deploy the web microservice on a WSGI-server. We chose to use the widely adopted Gunicorn as our WSGI server. This deployment is then virtualized using Docker, to achieve easy and quick deployments and maintenance. This also detaches the application environment from the host, which makes the application set up and dependency installation straight forward. We do not expose the Docker straight to the public internet, but rather use an NginX web-server as a reverse proxy to reroute incoming traffic to a localhost port in which the Python application is listening.

Authentication is currently done on a simple HTTP authentication, which is a feature provided by the NginX web-server.

4. EVALUATION

While evaluating the project, we were interested of the Medidash dashboard's overall ease of use. Parkinson's Disease patients are quite often elderly people [11], thus the interface of the dashboard was specially designed with their limitations in mind. Usability evaluation is often called as "expert evaluation", because expert in usability perform them. Usability evaluation is not just usability testing. In fact, usability testing is usually the last thing you do in a usability program. One might be eager to jump into testing right away, asking a question "why just not to do usability testing?" However, a question "what kind of usability improvement program is appropriate for our needs?" should be asked. Products have multifarious aspects, including (1) multiple languages in every product screens, (2) error messages and help messages, (3) tutorials, (4) documentation and user's guides and a lot more, when considered the product in context. The user audience of the product can be very broad and there are great amount of variations what the users will do with the product. Users may be first-time users or experienced users, so the individual user experience will differ from that of others. It is obvious that testing all these aspects is most likely impossible.

Usability tests lets us quantify to which extent the product meets the needs of its intended users. Usability testing gives us a target user's point of view and the tests should consist of tasks designed to represent real user cases. Most testing focuses on specific aspects of the product, rather than on all aspects of it. Usability tests and usability evaluation has the same goal: to improve the usability of a product.[59] As our software's (main) target audience is rather limited and the scale of the project is not very large, we decided to do a usability test, rather than usability evaluation. The participants should be as close as possible to the target group for accurate testing results. This means it would be ideal to test the dashboard with different sets of users. Dashboard will be used by Parkinson's Disease patients but also by their relatives and doctors. According to Nielsen [23], best testing results come with three to five users and running the tests as small as possible. Therefore, we aim to keep the test sessions short and concise, and the number of participants to near five. Optimally, the participants are males and females of different ages but due to the limited time we decided to do the tests at university premises. That's why we focus only on one group at the time to maintain the reliability of the tests.

4.1. Evaluation Plan

The tests consist of two loosely defined parts, which both contain a small set of tasks or questions. The first part deals with the dashboard's "Regular view" and the latter part deals with "Advanced view". The decision, how we will test the dashboard and how we will interpret the test results, created a lot of discussion. The system usability survey (SUS) was considered to be done after the tasks, allowing us to calculate the SUS score and analyze it. We would have also included open questions to give us more in-depth qualitative feedback about the benefits and shortcoming of the system. Even though SUS has become an industry standard and its results are reliable with small sample sizes, we decided not to use SUS because we were more interested in qualitative rather than quantitative feedback. Also, the scoring system of SUS is unnecessarily complex for this use case. The scores could deceive one to think that the score should be interpreted as percentages because the score is in scale from 0 to 100.[60] We decided that the participants will perform the tests in small groups and the feedback was given verbally during and after the tests.

While doing the tests the participants are encouraged to think aloud and discuss about the dashboard and the given tasks. We also included open questions to the end of the tests. The participants can talk about the questions with us or with themselves. The tests will be performed in an isolated space, the tests will be recorded and the actual tests are performed with computers. The tasks are small questions such as: “Describe what you see”, “What kind of information can be obtained from here?”, “How has the score developed during the last 30 days?” and “What has the intake been like during the past week?” Because the dashboard is informative but a rather small application, all the tasks should be pretty quick to perform. The tasks do survey the usability because they guide you to answer to questions what the actual users would be looking for. The open questions ask about what was good or bad, was something unclear and how could the dashboard be better. All of the evaluation sessions are to be recorded, to enable later analysis.

4.2. Evaluation Process

The evaluation consisted of three persons. There was a little brief at the beginning of each of the evaluation sessions. The brief gave the groups some context, like that the application is for visualization of medical measurements results, the data in the actual context would be from Parkinson’s disease patients and the data is collected with the STOP application. They were also told that the data used to draw the graphs was generated from their own measurements and would by default not be visible to anybody else. The provided test data to use the application was generated by us and consisted of roughly two weeks of regular data. In a real use case, this would not be very much. Overall impressions of the application seemed to be quite good. According to the test users, the user interface was clear and the site was easy to navigate. However, the plotly applications caused some problems as people miss clicked them and zoomed unintentionally. Especially in the advanced view was a lot of data and tables, so some tooltips would be a great assistance there.

4.2.1. Regular View

After the short briefing in the beginning of the session, the groups were free to look at the regular view and describe what they saw. In case the participants failed to describe the user interface voluntarily, we lead their span of attention with questions described in chapter 4.1. Overall, the main point of the regular view was very clear. The view itself was easy to look at as there were not any confusing elements. The view gathered comments as “nicely visualized” and “informative”. The test groups associated the smiley faces as intended; green for good, yellow for okayish and red for bad. Even though the feedback and the comments in the regular view was mainly positive, also a few of concerns sprung up. As we visualize the score for the last 30 days, last week and last day, there could be a possibility for patient to misunderstand the results. If a patient's disease is getting worse and she or he gets a good score one day, the patient could be lead to think his or her symptoms are getting better. The last day is not statistically significant compared to the similar values of last week or to the last 30 days.

Also the score limits for a smiley faces were noticed. If the score limit for the green smiley face is hard coded the patient will never get a positive feedback as the disease progresses. It would be important to scale the feedback for each patients personal needs and profile, rather than a comparison to a healthy individual. As STOP records data about the taken medication, it was mentioned that a similar indicator as with the scores would be nice. Or even a quick little reminder to take the pills would be useful.

4.2.2. Advanced View

If the participants did not move to the advanced view voluntarily, they were advised to do so after we were done with the regular view. Again, the groups were free to take look at the view and describe aloud what they saw. As the advanced view is primarily designed to be used by doctors and other professionals, there is considerably more information compared to the regular view. This was seen as groups explored the view longer in silence and could also be observed as visible confusion in their facial expressions. One of the questions asked would be translated from Finnish as "How would you evaluate the health status changes during the measurement period?" The questions caused silence, terms like "I believe", "I think" and overall difficulties to answer the question. After a while the groups did manage to answer the question and it was not so hard as it had seemed at first. It was agreed that the small amount of data made answering the question harder but the application seems to be able to visualize the temporal development of the scores.

With the guidance of the questions we discovered that the score was easy to follow from the graph. The graph also got some comments on how easy it would be to see how a new medication or new dosage amount affects the score. This raises a question whether patients would start to change their own doses of medication. The trend line of the score is calculated from overall score rather than calculating the trend from the visible score. This is also a possible source of confusion. The calendar element was found to be confusing. The calendar did not have any date bar and there was no information what month it was displaying. The date was inside the day, followed by the amount of medication taken. For example: "8, 2 meds", 8 meaning the day and the "2 meds" meaning medication has been taken twice during that day. The comma was found to be very confusing as the groups thought it meant there had been taken 8.2 medications. The calendar features colour coding for the amount of medication taken. It was found to be confusing as two medications was orange and four medications was red, as if taking less medications would be better.

Even with the small problems the advanced mode was found to be possibly good for the doctors. With the data visualization the doctor would be able to observe the progress of the Parkinson's disease and to reduce the control visits of the patient. However though, there was a lot of data and most of it was very mathematical. The view has a desperate need for hints and tooltips for not to be confusing at the first glance.

4.3. Evaluation Results

Overall, the evaluation phase was a success. Many of the key strengths and weaknesses of the application were addressed during the phase. Although our test users were not representative of the actual planned test users, they were able to provide us with valuable insight. This could have been enhanced even further if we had had Parkinson's disease patients and their doctors as test users as well as the students we already had. A sample size of 3 different users was used to evaluate the application. Now this sample size is not big enough for us to conduct a quantitative analysis on the application. It is, however, big enough to get a proper qualitative view of how the application is received. The main weaknesses that were identified were the lack of explanatory information and the confusing interface elements, especially the calendar. The things that were praised by all of the test subjects were the simplicity of the application, easy and fluent navigation and crisp and modern feel of the user interface.

The feedback and insight gained in the evaluation phase does allow further development to continue. In the future, we would like to continue on improving different aspects of the user interface. These development areas revolve around streamlining difficult to understand graphs and numerical information. Another important key aspect to improve are different hints and tooltips to enhance the user experience. Also, we might have had slightly wrong mindset when developing the user interface. Even though we knew and discussed about the actual users who are going to use the application, some of the decisions that were made were better suited for an engineer or an ICT-specialist rather than an elderly patient. We had clearly defined the two use cases, which were that the regular view was made for the patients and advanced view was made for their doctors. These made perfect sense for us, but were not mentioned within the application. This division of use cases is something that should be clarified in the tooltips that we planned to implement based on the evaluation feedback.

5. CONCLUSION AND FUTURE WORK

All in all, the project was a success. However, there are a few things that could and should be improved upon. One of the things, that we had planned on implementing but had to leave out due to technical difficulties and lack of time is the QR-log in system. Currently, as the log in feature does not support easy scalability for multiple users, the application is more of a proof of a concept rather than a fully fledged and production ready system. In further development the log in should be implemented through STOP application with dynamically created QR-codes as described in section 3.2.4. Things that should be improved, based on the evaluation phase are mainly in the user interface. These improvements include the clarification of the UI visuals and better informatics and hints for the plots.

A feature in the user interface that should be completely reworked is the medication calendar. It could be styled so that the colors are more obvious, for example ranging from green to red or from white to black instead of from red to orange and then yellow. Also, the numbering of the said calendar should be improved so that the number of the day is separated from the medication amount.

The effect of medication intake on the game score could be taken into account in the visual representations of the clinical picture. Off/on phases, as well as individual progression of the disease could make defining the relation between medication and games scores complicated, but it would be something to look into. However, if this was to be implemented, Medidash should inform the user that the taken medications have an effect on the score. For example, if the user has not taken medications and the score is rather low, Medidash would tell that there are no medication records and your score is low. If the score was high and there were no medication record, Medidash could remind to enter the taken medication to the STOP application. And most importantly, Medidash should notice and notify whether the current medication dosage helps alleviate the symptoms of the disease.

When these changes we have presented are implemented, the verification and evaluation process of the application should be extended to actual users. During the project we did not have a meeting with any kind of a Parkinson's disease specialist, and all the evaluation participants were university students. Although the evaluation process provided us with valuable insight even with non-optimal test users, the same evaluation should be conducted with Parkinson's disease patients and doctors.

The evaluation with real end users, especially with doctors, would give us a better idea about what data is actually important. For now it is uncertain what kind of data the doctors want to see and why. This brings up questions like how to manipulate the existing data to provide meaningful information. The fulfillment of these needs could require changes to the implementation of Medidash, or the implementation of the STOP application.

A proper and exhaustive data analysis in conjunction with a doctor would have to be done from already collected data to evaluate what information and in what form would be important for the users to see and follow. The data analysis could also find shortcomings like missing values and how the visualizations should be adjusted accordingly—as users might forget to play the game every day. Hidden patterns like cyclic events could be discovered and special visualizations could be built around them.

If Medidash was to be evaluated further with Parkinson's disease patients and specialists, the gained information could be useful in the development of other applications. From the data doctor could easily see the progression of the disease and assign a control visit or prescribe new amount of medicine. With enough data machine learning algorithm could be used to automatically observe progression of each patient's clinical picture. This way doctors should observe the patient's progression only when necessary.

More research of self medical journals on smart/mobile devices should be made, as we found barely any information on such topics. Questions like "how reliable are self medication journals on smart devices and how to make them more reliable?" are still without answers. The research would make it easier to research more about data visualization for users with limited capabilities and create new possibilities to develop a more reliable and versatile dashboard application. This would not only benefit applications made for Parkinson's disease patients, but applications made for any long term disease patients.

With such research, combined with machine learning, long term patients' could get information faster on the status of their diseases. Medication dosages could be evaluated easier without control visits. This could make patients' life easier, especially in rural areas where hospitals could be difficult to access. As Parkinson's disease is more common within the older age groups, the patients could be constrained by their ability to travel. This combined with that elderly people can often live in small rural towns would be cheaper for municipalities as they would not need to arrange transportation as often for them.

If Medidash was to be used in an actual production environment with real users using it daily, more work should be put into the security aspects. Currently, the login is safe but not very scalable. When processing sensitive or confidential data, security and managing possible risks should be of highest priority. Security issues such as SQL injections, DOM XSS and reflected XSS attacks, should be reviewed. Other possible risks include that users could make their own conclusions on the medication dosages. If machine learning algorithms were used within the application, extensive verification should be done to ensure their correctness. If these medication machine algorithms were to produce erroneous results, there could be fatal risks regarding the well-being of the patients.

6. REFERENCES

[1] Swan M. (2013) The quantified self: Fundamental disruption in big data science and biological discovery. Big Data 1, pp. 85–99.

[2] Swan M. (2012) Health 2050: The realization of personalized medicine through crowdsourcing, the quantified self, and the participatory biocitizen. Journal of personalized medicine 2, pp. 93–118.

[3] Swan M. (2009) Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. International journal of environmental research and public health 6, pp. 492–525.

[4] Rind A., Aigner W., Miksch S., Wiltner S., Pohl M., Turic T. & Drexler F. (2011) Visual exploration of time-oriented patient data for chronic diseases: Design study and evaluation, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7058 LNCS. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-82155197304&doi=10.1007%2f978-3-642-25364-5_22&partnerID=40&md5=092ba0a4c766f6b2d5ca07fe2aa2689f`, cited By :11.

[5] Rind A., Wang T.D., Aigner W., Miksch S., Wongsuphasawat K., Plaisant C. & Shneiderman B. (2011) Interactive information visualization to explore and query electronic health records. Foundations and Trends in Human-Computer Interaction 5, pp. 207–298. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84877897011&doi=10.1561%2f1100000039&partnerID=40&md5=3985ee136fbef9ab2db160b3165f14f6`, cited By :84.

[6] Wicks P., Massagli M., Frost J., Brownstein C., Okun S., Vaughan T., Bradley R. & Heywood J. (2010) Sharing health data for better outcomes on patientslikeme. Journal of medical Internet research 12, p. e19. URL: `https://www.ncbi.nlm.nih.gov/pubmed/20542858` `https://www.ncbi.nlm.nih.gov/pmc/PMC2956230/`, undefined; J1: J Med Internet Res.

[7] Jusufi I., Nyholm D. & Memedi M. (2014) Visualization of spiral drawing data of patients with parkinson's disease. In: Information Visualisation (IV), 2014 18th International Conference on, IEEE, pp. 346–350.

[8] Lo G., Suresh A.R., Stocco L., González-Valenzuela S. & Leung V.C. (2011) A wireless sensor system for motion analysis of parkinson's disease patients. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on, IEEE, pp. 372–375.

[9] Page A., Soyata T., Couderc J.P. & Aktas M. (2015) An open source ecg clock generator for visualization of long-term cardiac monitoring

data. IEEE Access 3, pp. 2704–2714. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84961299423&doi=10.1109%2fACCESS.2015.2509426&partnerID=40&md5=3d5d2f32aa698f1563bc899cf70c869d, cited By :14.

[10] Massano J. & Bhatia K.P. (2012) Clinical approach to parkinson's disease: features, diagnosis, and principles of management. Cold Spring Harbor perspectives in medicine , p. a008870.

[11] Jankovic J. (2008) Parkinson's disease: clinical features and diagnosis. J Neurol Neurosurg Psychiatry 79, p. 368. URL: http://jnnp.bmj.com/content/79/4/368.abstract.

[12] Association E.P.D. (2017), Mild memory and cognitive problems. URL: https://www.epda.eu.com/about-parkinsons/symptoms/non-motor-symptoms/mild-memory-and-cognitive-problems/.

[13] framework A., Aware framework context. URL: https://awareframework.com/context/.

[14] Farmer K.C. (1999) Methods for measuring and monitoring medication regimen adherence in clinical trials and clinical practice. Clinical therapeutics 21, pp. 1074–1090.

[15] Mahmoudi S.E., Akhondi-Asl A., Rahmani R., Faghih-Roohi S., Taimouri V., Sabouri A. & Soltanian-Zadeh H. (2010) Web-based interactive 2d/3d medical image processing and visualization software. Computer methods and programs in biomedicine 98, pp. 172–182.

[16] Boll F. & Brune P. (2015) User interfaces with a touch of grey?–towards a specific ui design for people in the transition age. Procedia Computer Science 63, pp. 511–516.

[17] Arias-Torres D. (2006) The design and evaluation of a pen-based computer interface for novice older users. In: 2006 Seventh Mexican International Conference on Computer Science, pp. 142–150. ID: 1.

[18] Kobayashi M., Hiyama A., Miura T., Asakawa C., Hirose M. & Ifukube T. (2011) Elderly user evaluation of mobile touchscreen interactions. In: IFIP Conference on Human-Computer Interaction, Springer, pp. 83–99.

[19] Nunes F., Silva P.A., Cevada J., Barros A.C. & Teixeira L. (2016) User interface design guidelines for smartphone applications for people with parkinson's disease. Universal Access in the Information Society 15, pp. 659–679. URL: https://doi.org/10.1007/s10209-015-0440-1, iD: Nunes2016.

[20] Jokela T., Iivari N. & Tornberg V. (2019) USING THE ISO 9241-11 DEFINITION OF USABILITY IN REQUIREMENTS DETERMINATION: CASE STUDIES.

[21] Nielsen J. (2003) Usability 101. Usability 101: Introduction to usability .

[22] Harley A. (2018), Ux expert reviews. URL: https://www.nngroup.com/articles/ux-expert-reviews/.

[23] Nielsen J. (1992) Finding usability problems through heuristic evaluation. pp. 373–380. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-57249090957&partnerID=40&md5=50320910f35b88dddb645fec785033cc, cited By :397; Export Date: 6 February 2019.

[24] Mustaniemi J. (2009) Käytettävyyden arviointimenetelmät .

[25] Laine A. (2004), Hahmolait käytettävyyden parantajina. URL: http://www.mit.jyu.fi/opetus/opinnayte/LuK/Hahmolait/.

[26] Sainio L. (2009) Ohjelmistotestauksen menetelmät ja työvälineet. Ph.D. thesis.

[27] Hangasmaa H. (2010) Käytettävyyden merkitys ja käytettävyystestaus. Ph.D. thesis.

[28] Iivonen S.P. (2012) Käyttöliittymän suunnitteluprosessi. Ph.D. thesis.

[29] Fielding R.T. & Taylor R.N. (2002) Principled design of the modern web architecture. ACM Transactions on Internet Technology (TOIT) 2, pp. 115–150.

[30] Fielding R. & Reschke J. (2014) Hypertext transfer protocol (http/1.1): Semantics and content. Hypertext transfer protocol (HTTP/1.1): Semantics and content .

[31] Pohja M. (2011) Web application user interface technologies. Ph.D. thesis. URL: http://urn.fi/URN:ISBN:978-952-60-4011-0[urn].

[32] Oracle (2019), Swing apis and developer guides. URL: https://docs.oracle.com/javase/8/docs/technotes/guides/swing/.

[33] Team T.G., About gtk+ project. URL: https://www.gtk.org/gtk-doc/.

[34] project T.Q. (2018), About qt. URL: https://wiki.qt.io/About_Qt.

[35] Inc A. (2017), Developer release notes cocoa application framework (macos 10.13). URL: https://developer.apple.com/library/archive/releasenotes/AppKit/RN-AppKit/index.html.

[36] Reeve A., Simcox E. & Turnbull D. (2014) Ageing and parkinson's disease: Why is advancing age the biggest risk factor? Ageing Research Reviews 14, pp. 19 – 30. URL: http://www.sciencedirect.com/science/article/pii/S1568163714000051.

[37] Aigner W., Miksch S., Müller W., Schumann H. & Tominski C. (2007) Visualizing time-oriented data-a systematic view. Computers and Graphics (Pergamon) 31, pp. 401–409. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-34248230618&doi=10.1016%2fj.cag.2007.01.030&partnerID=40&md5=90c3a020f109e220e41cb6fdac9a283b, cited By :169.

[38] Twitter search tweets rest api overview. URL: https://developer.twitter.com/en/docs/tweets/search/overview.

[39] LLC W.A. (2019), Wolfram alpha api's. URL: https://products.wolframalpha.com/api/.

[40] Rest interfaces of the city of oulu. URL: https://data.ouka.fi/data/fi/dataset?res_format=REST.

[41] Youtube developer documentation. URL: https://developers.google.com/youtube/documentation/.

[42] Rodriguez A. (2008) Restful web services: The basics. IBM developerWorks 33.

[43] Berners-Lee T., Fielding R. & Masinter L. (2004) Uniform resource identifier (uri): Generic syntax. Tech. rep.

[44] Team P., Flask foreword. URL: http://flask.pocoo.org/docs/1.0/foreword/.

[45] Plotly, Plotly.js. URL: https://plot.ly/javascript/.

[46] Inc. F., Reactjs - front. URL: https://reactjs.org.

[47] dash, dash introduction. URL: https://dash.plot.ly/introduction.

[48] plotly, dash architecture. URL: https://medium.com/@plotlygraphs/introducing-dash-5ecf7191b503.

[49] inc. F., Reactjs - dom. URL: https://reactjs.org/docs/faq-internals.html.

[50] Team P., Flask - extensions. URL: http://flask.pocoo.org/extensions/.

[51] Cohen S., flask - pinterest. URL: https://www.quora.com/What-challenges-has-Pinterest-encountered-with-Flask/answer/Steve-Cohen?srid=hXZd&share=1.

[52] PyCon . & Sanders R., Flask - linkedin. URL: https://www.youtube.com/watch?v=OXN3wuHUBP0#t=46.

[53] Dubey P., An introduction to plotly.js. an open source graphing library. URL: https://medium.freecodecamp.org/an-introduction-to-plotly-js-an-open-source-graphing-library-c036a1876e2e.

[54] Plotly, Plotly open source announcement. URL: https://plot.ly/javascript/open-source-announcement/.

[55] Walt S.V.D., Colbert S.C. & Varoquaux G. (2011) The numpy array: a structure for efficient numerical computation. Computing in Science & Engineering 13, p. 22.

[56] pandas (2019), pandas: powerful python data analysis toolkit. URL: https://github.com/pandas-dev/pandas.

[57] pandas (2019), Python data analysis library. URL: https://pandas.pydata.org/index.html.

[58] Pereira R. (2016), How to implement a mechanism like qr code scanning of whatsapp web. URL: http://blog.jbaysolutions.com/2016/09/20/how-to-implement-a-mechanism-like-qr-code-scanning-of-whatsapp-web/.

[59] Rosenbaum S. (1989) Usability evaluations versus usability testing: when and why? IEEE Transactions on Professional Communication 32, pp. 210–216. ID: 1.

[60] Brooke J. (1996) Sus-a quick and dirty usability scale. Usability evaluation in industry 189, pp. 4–7.